Covering Polygons with Rectangles

Roland Glück*

Abstract

A well-known and well-investigated family of hard optimization problems concerns variants of the cutting stock or nesting problem, i.e. the non-overlapping placing of polygons to be cut from a rectangle or the plane whilst minimizing the waste. Here we consider an in some sense inverse problem. Concretely, given a set of polygons in the plane, we seek the minimum number of rectangles of a given shape such that every polygon is covered by at least one rectangle. As motions of the given rectangle we investigate the cases of translation and of translation combined with rotation.

1 Introduction

In manufacturing, one often faces the problem of cutting a set of given polygons out of a piece of material (e.g. sheet metal or cloth) in a way which produces as less waste as possible. In this paper we investigate the subsequent step in production technology: once the pieces are cut out they will be picked off and transported by a suitable device. Here we restrict ourselves to a rectangular gripper and various degrees of freedom: The first case is a rectangular gripper which can be translated both in x- and y-direction; the second case deals with a rectangular gripper which additionally has the possibility of being rotated. The concrete motivation of this paper is a machine which cuts polygons out of a carbon fiber fabric and grasps the cut pieces with a rectangular gipper with vacuum suction devices.

Basically, this task corresponds to covering a set of polygons by copies of a rectangle such that every polygon is contained in at least one rectangle. There is a lot of work about covering sets of points with rectangles as in [4, 6] but none of them matches our problem. Due to the NP-hardness of all these problems (see [5] for a comprehensive list) we suspect that the problems we consider are also NP-hard. We do not propose an approximation algorithm but a family of exact algorithms which works well on practical instances.

The paper is organized as follows: Section 2 provides definitions and states the problem in a generic way. In Section 3 we prove some useful lemmata for the further course. A generic approach to the problems is presented in Section 4, while Section 5 deals with some implementation issues and provides experimental results. The finishing Section 6 gives a short summary and directions of future work.

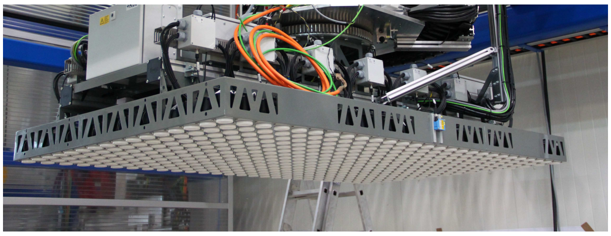

Figure 1: The bottom side of the gripper

2 Definitions

In order to formalize our task we introduce the concept of a packing: a *packing* $\mathbf{P} = \{P_1, P_2, \ldots, P_n\}$ is a set of n possibly overlapping simple polygons P_1, P_2, …, P_n. Clearly, $|\mathbf{P}|$ denotes the number of polygons of $\mathbf{P}$, and we use the notation $\|\mathbf{P}\|$ for the overall number of vertices in $\mathbf{P}$. We say that a packing $\mathbf{P}$ is *covered* by a set $\mathbf{C} = \{R_1, R_2, \ldots, R_m\}$ of rectangles if each polygon of $\mathbf{P}$ is contained in at least one rectangle of $\mathbf{C}$. If a rectangle R' arises from a rectangle R by a translation we call R' a *translation* of R, and if R' arises from R by a translation and a rotation we say that R' is a *general motion* of R (this is equivalent to the term "rigid motion" in [2]). With these namings we can define the main theme of our investigations:

Definition 1 *Let* $\mathbf{P}$ *be a packing and* R *an axis-aligned rectangle (the so-called gripper). We call a set of rectangles* $\mathbf{C} = \{R_1, R_2, \ldots, R_m\}$ *a* translational (general) cover *of* $\mathbf{P}$ *if* $\mathbf{C}$ *covers* $\mathbf{P}$ *and all rectangles of* $\mathbf{C}$ *are translations (general motions) of* R.

Since we are interested in covering a packing with as few as possible rectangles we call a cover of every kind *optimal* if it has minimal cardinality amongst all covers of the respective kind. To ease wording we refer by the term *cover* to both a translational or general cover. For a packing $\mathbf{P}$ and a rectangle R we denote the set of polygons of $\mathbf{P}$ covered by R by $\mathsf{cov}(R, \mathbf{P})$. We extend this notion to a set $\mathbf{R}$ of rectangles by $\mathsf{cov}(\mathbf{R}, \mathbf{P}) := \bigcup_{R \in \mathbf{R}} \mathsf{cov}(R, \mathbf{P})$.

*German Aerospace Center roland.glueck@dlr.de

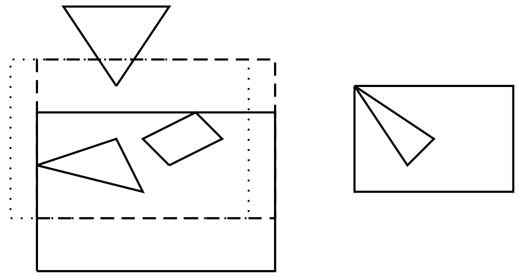

Figure 2: Translational Alignments

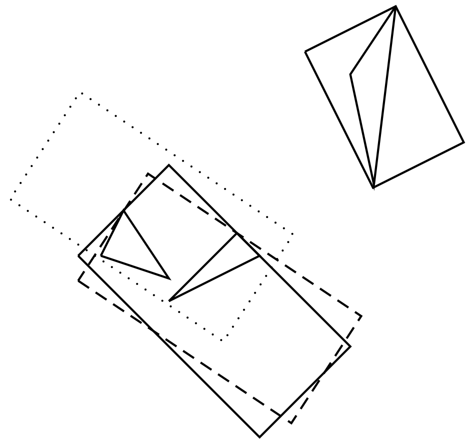

Figure 3: General Motion Alignments

3 Basic Facts and Observations

In general, the set of covers of a given packing is uncountable so we will have to discretize the search space in a suitable manner. As tools for reducing the number of coverings to consider for computation we state some useful properties and lemmata. The first lemma which holds for both translational and general covers will pave the way for a recursive approach to our problem:

Lemma 1 *Let* $\mathbf{P}$ *be a packing,* $\mathbf{C}$ *an optimal cover of* $\mathbf{P}$ *and* R_j *an arbitrary rectangle of* $\mathbf{C}$*. Then* $\mathbf{C}\backslash\{R_j\}$ *is an optimal cover of* $\mathbf{P}\backslash\mathsf{cov}(R_j, \mathbf{P})$*.*

Proof. Assume there is a cover $\mathbf{C}'$ of $\mathbf{P}\backslash\mathsf{cov}(R_j, \mathbf{P})$ with $|\mathbf{C}'| < |\mathbf{C}| - 1$. Then $\mathbf{C}' \cup \{R_j\}$ is a cover of $\mathbf{P}$ with a size of $|\mathbf{C}'| + 1 < |\mathbf{C}|$ which contradicts the optimality of $\mathbf{C}$. ☐

In the next lemma we give a first step towards discretization of the search space in the case of a translational cover:

Lemma 2 *Let* $\mathbf{P}$ *be a packing and* $\mathbf{C} = \{R_1, R_2, \ldots, R_m\}$ *an optimal translational cover of* $\mathbf{P}$*. Then there are points* p_1 *and* p_2 *of* $\mathbf{P}$ *and an index* j *together with an axis-aligned rectangle* R'_j *fulfilling the following properties:*

1. p_1 *has minimal x-coordinate amongst all points of* $\mathbf{P}$*,*

2. p_1 *lies on the left side of* R'_j*,*

3. p_2 *lies on the upper side of* R'_j*,*

4. *there are polygons* $P_1, P_2 \in \mathbf{P}$ *such that for* $i \in \{1, 2\}$ p_i *is a vertex of* P_i *and* P_i *is contained in* R'_j*, and*

5. $\mathbf{C}\backslash\{R_j\} \cup \{R'_j\}$ *is an optimal translational cover of* $\mathbf{P}$*.*

Note that we do not require that p_1 and p_2, P_1 and P_2 as well as R_j and R'_j are distinct. Moreover, if p_1 and p_2 are equal then they coincide with the upper left vertex of R'_j.

Proof. Let p_1 be a point of $\mathbf{P}$ with minimal x-coordinate and P_1 a polygon of $\mathbf{P}$ which has p_1 as a vertex. Then there is a rectangle $R_j \in \mathbf{C}$ containing P_1. Now we translate R_j in positive x-direction till p_1 lies on the left side of the translated rectangle $\hat{R}_j$. Clearly, we have $\mathsf{cov}(\hat{R}_j, \mathbf{P}) \supseteq \mathsf{cov}(R_j, \mathbf{P})$. Subsequently, we translate $\hat{R}_j$ in negative y-direction till a point p_2 with the following properties lies on the upper side of the translated rectangle R'_j:

1. All polygons of $\mathbf{P}$ with p_2 as a vertex contained in $\hat{R}_j$ are contained in R'_j, and

2. p_2 is a point with maximal y-coordinate fulfilling the above requirements.

Then we have $\mathsf{cov}(R'_j, \mathbf{P}) \supseteq \mathsf{cov}(\hat{R}_j, \mathbf{P}) \supseteq \mathsf{cov}(R_j, \mathbf{P})$, so $\mathbf{C}\backslash\{R_j\} \cup \{R'_j\}$ is indeed an optimal translational cover of $\mathbf{P}$. Moreover, p_1, p_2 and P_1 meet their requirements by construction, and for P_2 we can choose an arbitrary polygon with p_2 as a vertex which is contained in R'_j. ☐

The general situation is depicted in the left part of Figure 2: R_j correponds to the dotted rectangle, $\hat{R}_j$ to the dashed one, and the final rectangle R'_j is drawn with a full line. A pathological example where p_1 and p_2 as well as P_1 and P_2 coincide can be seen in the right part of the same figure.

A similar property can be stated for general covers (this and the previous lemma show some similarity to the term "stable placement" in[1]):

Lemma 3 *Let* $\mathbf{P}$ *be a packing and* $\mathbf{C}$ *an optimal general cover of* $\mathbf{P}$*. Then for every polygon* $P_{pi} \in \mathbf{P}$ *there are points* p_1*,* p_2 *and* p_3 *of* $\mathbf{P}$ *and an index* j *together with a rectangle* R'_j *fulfilling the following properties:*

1. *R'_j contains P_{pi},*
2. *p_1 and p_2 are distinct and lie on two different adjacent sides of R'_j*
3. *p_1, p_2 and p_3 lie on sides of R'_j,*
4. *there are polygons $P_1, P_2, P_3 \in \mathbf{P}$ such that for $i \in \{1, 2, 3\}$ p_i is a vertex of P_i and P_i is contained in R'_j,*
5. *$\mathbf{C}\backslash\{R_j\} \cup \{R'_j\}$ is an optimal general cover of $\mathbf{P}$.*

Proof. Let $R_j \in \mathbf{C}$ be a rectangle containing P_{pi}. We apply to R_j similar translations as in Lemma 2 but do not translate in x- and y-direction but in directions parallel to adjacent sides of R_j. Doing so, we end up with a rectangle $\hat{R}_j$ and two (not necessarily distinct!) points p_1 and p_2 with the following properties:

1. $\hat{R}_j$ contains P_{pi},
2. p_1 and p_2 lie on adjacent sides of $\hat{R}_j$,
3. $\mathsf{cov}(\hat{R}_j, \mathbf{P}) \supseteq \mathsf{cov}(R_j, \mathbf{P})$, and
4. there are polygons $P_1, P_2 \in \mathbf{P}$ such that for $i \in \{1, 2\}$ p_i is a vertex of P_i and P_i is contained in $\hat{R}_j$.

Now we perform a general motion of $\hat{R}_j$ combined of a clockwise rotation and suitable translation which keeps p_1 and p_2 on their respective sides. There are two cases:

1. p_1 and p_2 coincide. Then the described general motion is a simple rotation of $\hat{R}_j$ around p_1. This rotation is continued as long as a point p_3 lies on a side of the resulting rectangle R'_j such that the following properties hold:
   (a) R'_j contains P_{pi},
   (b) there are polygons $P_1, P_3 \in \mathbf{P}$ such that for $i \in \{1, 3\}$ p_i is a vertex of P_i and P_i is contained in R'_j, and
   (c) $\mathsf{cov}(R'_j, \mathbf{P}) \supseteq \mathsf{cov}(\hat{R}_j, \mathbf{P})$.
2. p_1 and p_2 are distinct. Here we continue the general motion till one of the following two cases concerning the arising rectangle R'_j occurs:
   (a) p_1 or p_2 coincide with a vertex of R'_j, or
   (b) there is a point on a side of R'_j such that
      i. R'_j contains P_{pi},
      ii. there are polygons $P_1, P_2, P_3 \in \mathbf{P}$ such that for $i \in \{1, 2, 3\}$ p_i is a vertex of P_i and P_i is contained in R'_j, and
      iii. $\mathsf{cov}(R'_j, \mathbf{P}) \supseteq \mathsf{cov}(\hat{R}_j, \mathbf{P})$.

Now, after possibly necessary renamings, p_1, p_2 and p_3 together with R'_j meet the requirements of the lemma. □

The general situation is shown in the lower left part of Figure 3: the dotted rectangle corresponds to R_j, the dashed one to $\hat{R}_j$ and the fully lined to R'_j. An extreme situation is illustrated by the upper right part of the same figure.

4 General Approach

We will now introduce an exact generic algorithm for the minimal cover problem. For the sequel we fix a rectangle R which we will use as gripper for a translational or general cover of a packing $\mathbf{P}$.

Let us assume we have an algorithm candidate_rectangles which determines for every packing $\mathbf{Q}$ a finite set of translations or general motions of R such that for every optimal cover $\mathbf{C}$ of $\mathbf{Q}$ and every $R_{cov} \in \mathbf{C}$ there is an $R_{cand} \in$ candidate_rectangles such that $\mathsf{cov}(R_{cov}, \mathbf{Q}) \subseteq \mathsf{cov}(R_{cand}, \mathbf{Q})$ holds. Together with a function simp_cov which computes an arbitrary cover (which can be done by packing each polygon into a rectangle of the gripper's shape) we can formulate the generic Algorithm 1 whose correctness is ensured by Lemma 1.

Algorithm 1 Generic Branch and Bound Algorithm

Require: A Packing $\mathbf{P}$ and a Gripper R
set<rectangle> *global_cover* = simp_cov($R, \mathbf{P}$)
int *global_depth* = |*global_cover*|
Branch and Bound(0,$\emptyset$)
Ensure: *global_cover* is an optimal cover of $\mathbf{P}$ by R with cardinality *global_depth*

function Branch and Bound(int *depth*, set<rectangle> *rectangles*)
  if *rectangles* covers $\mathbf{P}$ **then**
    global_depth = *depth*
    global_cover = *rectangles*
  else if *depth* < *global_depth* **then**
    set<rectangle> *candidate_rectangles* =
    candidate_rectangles($\mathbf{P}\backslash$cov(*rectangles*, $\mathbf{P}$))
    for all $R \in$ *candidate_rectangles* **do**
      Branch and Bound(*depth* + 1, *rectangles* $\cup \{R\}$)
    end for
  end if
end function

Depending on whether one is interested in an optimal translational or general cover the function candidate_rectangles has to be implemented in different ways. Some possibilities are described in the next section.

Algorithm 1 is an exact algorithm so we cannot expect a polynomial running time. In the worst case, there are $\mathcal{O}(|\mathbf{P}|^{2|\mathbf{P}|})$ calls of Branch and Bound. As we will see in the next section, the candidate rectangles can be computed in $\mathcal{O}(\|\mathbf{P}\|)$ time for the translational and in $\mathcal{O}(\|\mathbf{P}\|)^3$ time for the general case.

5 Implementation Sketch and Experimental Results

A short look at Algorithm 1 reveals that a BFS in the induced search graph will lead to a faster implementation. As usual, the drawback of this approach is a greater amount of space required during the computation. We implemented both versions and observed that the BFS approach fits our practical problems better.

Lemmata 2 and 3 provide methods for computing the set *canditate_rectangles* in Line 9 of Algorithm 1.

In the case of a translational cover we observe that a translation of a rectangle is uniquely determined by the position of its upper left vertex. Moreover, given two distinct points, there at most one translations which make the two points lie on adjacent sides according to Lemma 2. In the sequel we will concentrate on the general cover problem because our gripper can also be rotated around the z-axis. Nevertheless, we implemented our algorithm for the translational cover and could solve instances with 25 polygons and 1250 vertices in less than a second on an Intel i7-4770 CPU with 3.4 GHz.

Similarly, each pair or triple of distinct point gives raise to only a finite number of general motions of given rectangle meeting the requirements from Lemma 3. So we iterate over the points or pairs or triples of points from the packing under consideration (concretely $\mathbf{P}\backslash\mathsf{cov}(rectangles, \mathbf{P})$ in Line 9 of Algorithm 1) and determine all motions of R which fulfill the conditions of Lemma 2 or 3, resp. Of course, it suffices to keep only those rectangles which cover a maximal set of polygons.

We refined this approach by the following idea: first, for every polygon P from the initial nesting, we generate a list $P_1^P, P_2^P, \ldots, P_m^P$ of compatible polygons which can be covered by the given gripper together with P. Second, we use these lists to compute the candidate rectangles for a packing $\mathbf{P}'$ arising during the execution as follows: we choose a pivot polygon P_{pi} from $\mathbf{P}'$ and iterate over all triples $(P_i^{P_{pi}}, P_j^{P_{pi}}, P_k^{P_{pi}})$ of compatible polygons of P_{pi} with $i \leq j \leq k$. For every such triple we compute the convex hull and determine for every tuple respectively triple of points of the convex hull the rectangles according to Lemma 3. As mentioned above, we only keep the rectangles covering a maximal set of polygons. The restriction to the points of the convex hull is justified by the fact that the covering rectangle has to contain all polygons of $\{P_i^{P_{pi}}, P_j^{P_{pi}}, P_k^{P_{pi}}\}$ which is equivalent to it that it contains the convex hull of these polygons. A crucial point is the choice of the polygon P_{pi} from Lemma 3 as pivot polygon in order to keep the branching degree of the algorithm at a low level. Experiments showed that a good choice for the pivot polygon is a polygon which has minimal distance to a vertex of an axis parallel rectangular minimal bounding box of $\mathbf{P}'$.

As one would expect, our experiments indicated a running time cubic in the overall number of vertices and roughly exponential in the number of polygons. Our implementation in Java solved instances from practice with 25 polygons and 1250 vertices in about 20 minutes on average in the same environment as above.

6 Conclusion and Outlook

Our algorithm seems to be applicable to practical instances. However, there is room for further improvements. In our setting the algorithm was run on only one core which is clearly not optimal since it is obviously easy to construct a parallelized version. Another idea is to compare other strategies than described above for finding the pivot polygon. From a theoretical point of view, it will be interesting to show the conjectured NP-completeness of our problems.

Acknowledgments

The author is grateful to Torben Hagerup, Christian Rähtz, Lev Sorokin and the anonymous reviewers for valuable hints and remarks.

References

[1] Chazelle, B.: The polygon containment problem. Advances in Computing Research, 1–33. JAI Press (1983)

[2] Dickerson, M., Scharstein, D.: Optimal placement of convex polygons to maximize point containment. Computationl Geometry 11(1), 1–16 (1998)

[3] Dowsland, K.A.: Determining an upper bound for a class of rectangular packing problems. Computers & OR 12(2), 201–205 (1985)

[4] Fowler, R.J., Paterson, M., Tanimoto, S.L.: Optimal packing and covering in the plane are np-complete. Inf. Process. Lett. 12(3), 133–137 (1981)

[5] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)

[6] Hochbaum, D.S., Maass, W.: Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM 32(1), 130–136 (1985)